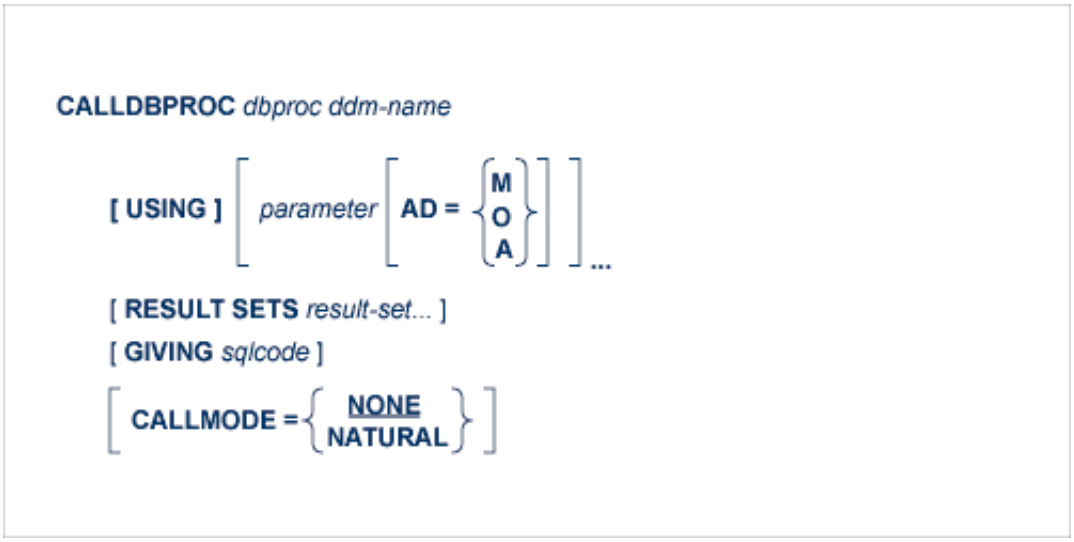
CALLDBPROC



Function

The statement CALLDBPROC is used to invoke a stored procedure of the SQL database system to which Natural is connected.

The stored procedure can be either a Natural subprogram or a program written in another programming language.

In addition to the passing of parameters between the invoking object and the stored procedure, CALLDBPROC supports "result sets"; these make it possible to return a larger amount of data from the stored procedure to the invoking object than would be possible via parameters.

The result sets are "temporary result tables" which are created by the stored procedure and which can be read and processed by the invoking object via a READ RESULT SET statement.

Note:
In general, the invoking of a stored procedure could be compared with the invoking of a Natural subprogram: when the CALLDBPROC statement is executed, control is passed to the stored procedure; after processing of the stored procedure, control is returned to the invoking object and processing continues with the statement following the CALLDBPROC statement.

dbproc

As dbproc you specify the name of the stored procedure to be invoked. The name can be specified either as an alphanumeric variable or as a constant (enclosed in apostrophes).

The name must adhere to the rules for stored procedure names of the target database system.

If the stored procedure is a Natural subprogram, the actual procedure name must not be longer than 8 characters.

ddm-name

The name of a DDM must be specified to provide the "address" of the database which executes the stored procedure. For more information see ddm-name.

parameter

As *parameter*, you can specify parameters which are passed from the invoking object to the stored procedure.

A *parameter* can be a host-variable (optionally with INDICATOR and LINDICATOR clauses), a constant, or the keyword NULL.

See further details on host-variable.

AD=

If the *parameter* is a *host-variable*, you can mark it as follows:

AD=O	Non-modifiable, see Session Parameter AD=O. (Corresponding procedure notation in DB2 for OS/390: IN.)
AD=M	Modifiable, see Session Parameter AD=M. (Corresponding procedure notation in DB2 for OS/390: INOUT.)
AD=A	For input only, see Session Parameter AD=A. (Corresponding procedure notation in DB2 for OS/390: OUT.)

If the *parameter* is a *constant*, AD cannot be explicitly specified. For constants AD=O always applies.

result-set

As result-set you specify a field in which a result-set locator is to be returned.

A result-set has to be a variable of format/length I4.

The value of a result-set variable is merely a number which identifies the result set and which can be referenced in a subsequent READ RESULT SET statement.

The sequence of the result-set values correspond to the sequence of the result sets returned by the stored procedure.

The contents of the result sets can be processed by a subsequent READ RESULT SET statement.

If no result set is returned, the corresponding result-set variable will contain "0".

On mainframe computers, multiple result sets can be specified. On all other platforms, only one result set can be specified.

GIVING sqlcode

This option may be used to obtain the SQL code of the SQL CALL statement invoking the stored procedure.

If this option is specified and the SQL code of the stored procedure is not "0", no Natural error message will be issued. In this case, the action to be taken in reaction to the SQL code value has to be coded in the invoking Natural object.

The sqlcode field has to be a variable of format/length I4.

If the GIVING sqlcode option is omitted, a Natural error message will be issued if the SQL code of the stored procedure is not "0".

CALLMODE

If the stored procedure is a Natural subprogram, CALLMODE=NATURAL has to be specified.

Note:
CALLMODE=NATURAL also has an impact on internal parameters that are passed to/from the stored procedure; see the corresponding Natural database interface documentation for details. CALLMODE=NATURAL is only available on mainframe computers.

Example

The following example shows a Natural program that calls the stored procedure ’demo_proc’ to retrieve all names of table PERSON that belong to a given range.

Three parameter fields are passed to ’demo_proc’: the first and second parameters pass starting and ending values of the range of names to the stored procedure, and the third parameter receives a name that meets the criterion.

In this example, the names are returned in a result set that is processed using the READ RESULT SET statement.

```
DEFINE DATA LOCAL
1 PERSON VIEW OF DEMO-PERSON
  2 PERSON_ID
  2 LAST_NAME
1 #BEGIN (A2) INIT <’AB’>
1 #END  (A2) INIT <’DE’>
1 #RESPONSE (I4)
1 #RESULT (I4)
1 #NAME (A20)
END-DEFINE

...

CALLDBPROC ’demo_proc’ DEMO-PERSON #BEGIN (AD=O) #END (AD=O) #NAME (AD=A)
    RESULT SETS #RESULT
    GIVING #RESPONSE

READ RESULT SET #RESULT INTO #NAME FROM DEMO-PERSON
    GIVING #RESPONSE
  DISPLAY #NAME
END-RESULT

...

END
```


See the corresponding Natural database interface documentation in the Natural for Mainframes documentation for further examples.

